SCALABILITY STUDY AND DISTRIBUTED SIMULATIONS OF AN ATM NETWORK MANAGEMENT SYSTEM BASED ON INTELLIGENT AGENTS*

Pere Vilà, Josep L. Marzo**
Institut d'Informàtica i Aplicacions (IIiA)
Universitat de Girona
Avda Lluís Santaló s/n, 17071 Girona (SPAIN)
phone: +34 972418475   fax: +34 972418098
e-mail: {perev | marzo}@eia.udg.es

KEYWORDS

Telecommunications, Network Management, Distributed Simulation, Multi-Agent Systems, ATM.

ABSTRACT

This paper examines the scalability problem of an Asynchronous Transfer Mode (ATM) network management system architecture. This architecture is based on distributed artificial intelligence schemes, concretely on a Multi-Agent system. The idea is to use this Multi-Agent system dynamically to manage the Virtual Path network, i.e. dynamic bandwidth allocation and fault restoration. The Multi-Agent system architecture, its evaluation and the simulations designed for this system are presented. The simulations are distributed systems using several processes to simulate each ATM node. The processes constitute a simulated ATM test-bed to which other processes and experiments can be added. For example, we use this platform to test our management system based on a Multi-Agent System (which is also a distributed system).

1 INTRODUCTION

There is a need for automated network management tools in large and complex networks, because human effort is insufficient. Traditional network management systems are centralised approaches (Figure 1), which introduce a scalability problem when the network grows. Typically, a network management system is a collection of tools for network monitoring and control that is integrated in a single operator interface with a powerful but user-friendly set of commands for performing most or all network management tasks.

Traditional architecture of a network management system is as follows. Each network node contains a collection of software devoted to the network management task, usually referred to as a network management entity (NME) or Agent. Each NME performs the following tasks:

- Collects statistics on communications and network-related activities
- Stores statistics locally
- Responds to commands from the network control centre, including commands to transmit collected statistics to the network control centre, change a parameter, provide status information, and generate artificial traffic to perform a test.

At least one host in the network is designed as the network control centre (NCC) or Manager. The Manager-Agent communication is carried out through an application-level network management protocol that uses the communications architecture in the same way as any other distributed application.

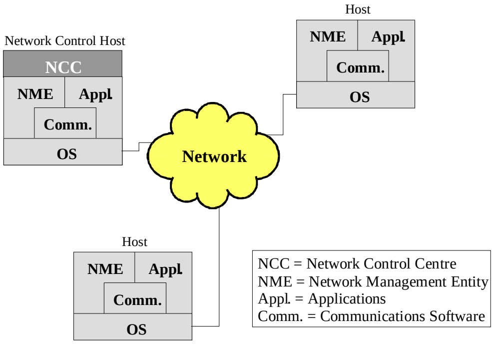

Figure 1: Centralised Network Management

Well-known management standards include Telecommunications Management Network (TMN) (Sidor 1998, Manley and Thomas 1997) and Simple Network Management Protocol (SNMP) (Stallings 1996, Stallings 1998). More information on network management standards can be found in (Black 1994, Raman 1998).

A network management system is scalable if its run-time performance does not degrade exponentially as both the network and the number of users expand. Centralised approaches are not scalable because when the network grows (i.e. more nodes and links are added) the number of messages from/to the NCC grows exponentially (due to the pooling loops and events) and the data that the NCC needs to process becomes unmanageable. Hierarchical Network Management Architectures can manage larger networks than pure centralised architectures, but when the network grows even more, they suffer the same scalability problem.

It is very difficult to ensure the scalability of a distributed system. There have been some attempts to design a method to evaluate scalability in a distributed system (Jogalekar and Woodside 1998), and even some specific ideas for the scalability evaluation of multi-agent systems (Lee et al. 1998), but the final evaluation of a concrete system design and the comparison with other designs is still very difficult.

Recent studies propose other solutions to these network management problems. Most of these studies agree that some management tasks cannot be done in a centralised manner after the network reaches a certain size, and that these management tasks should be automated and distributed into the network.

---

* This study was partially supported by the CICYT (Spanish Education Ministry, under contracts TEC-98-0408-C02-01 and TEL-99-0976)
** Senior visiting scientist at the International Computer Institute (ICSI), University of California, Berkeley.

Some different schemes for this kind of automation have been developed, i.e. Management by Delegation (MbD) techniques (Yemini et al. 1991, Goldszmidt and Yemini 1998, Betser 1996) and Distributed Management techniques, e.g. Distributed Bandwidth Management (Larsson and Arvidsson 1997, Larsson and Arvidsson 1999). These techniques are usually applied only to a small and specific number of management tasks. This is because both the MbD and Distributed techniques entail two difficult problems.

First, even though management activities can be performed in a distributed way, human managers still demand a “centralised” management view. Second, some management functions are very difficult to distribute, because there are certain procedures that need an overall view of the network status.

These problems are usually solved by using some centralised mechanisms alongside distributed mechanisms (the scalability problem arises) or by maintaining some kind of replication of the NCC and management databases (the duplication maintenance problem arises).

Some techniques to overcome these two drawbacks have appeared. They seek to tackle network management in different terms. For the day-by-day network management operation, the use of intelligent software agents is proposed. These are defined as software entities with special properties (autonomy, social ability, reactivity and pro-activeness) (Wooldridge and Jennings 1995, Nwana 1996). The major objective is to automate these management functions and so improve network performance. They tackle the problems mentioned as follows. The problem of giving a whole network status view to one human network manager is usually solved without any human involvement at all. In other words, the intelligent agents have the ultimate responsibility for network management: all the management functions are automated. The idea is to hide the management complexity from the human network managers and to give them a not-real-time picture of network status, but a picture which is not out of date, and with which they can introduce goals, requirements, restrictions, etc. into the intelligent agent system. The difficult-to-distribute functions (the second problem) are usually implemented by using various techniques and tools from the Distributed Artificial Intelligence field such as negotiating, planning and collaborating techniques.

As network management is a good field for applying these techniques, several Multi-Agent approaches to these various network management problems have been developed (Bigham et al. 1999a, Bigham et al. 1999b, Davison et al. 1998, Hardwicke and Davison 1998, Hayzelden and Bigham 1998, Hayzelden et al. 1999, Somers 1996, Somers et al. 1997, Lucent Technologies 1997). Hayzelden and Bigham 1999 give an overview of intelligent agents in telecommunications. Their approach is quite close to our own, which tackles, broadly speaking, dynamic Virtual Path (VP) management in ATM networks. In the following introductory point the problems that our Multi-Agent architecture confronts are defined. The following section briefly describes the system architecture and the main objectives. Section 4 presents a scalability study of the proposed architecture and Section 5 specifies the restrictions and requirements, and presents the planned distributed simulations. Finally, conclusions and future work is presented.

2 PROBLEM SPECIFICATION

ATM networks have been designed to support a wide range of services with diverse characteristics (Kyas 1995, Le Boudec 1992, Sykas et al. 1991). They have several layers of hierarchy. One of these layers is the Virtual Path layer, which is used to simplify the establishment of new connections and also constitutes a virtual topology over the physical network (Figure 2). This allows dynamic management of this virtual topology and its adaption to improve network resource use (Friesen et al. 1996, Sato et al. 1990). The three main VP management functions our approach seeks to tackle, bandwidth management, fault restoration and spare capacity planning, are described now.

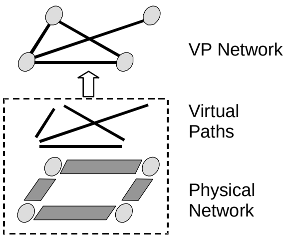

Figure 2: Virtual Path Network

Bandwidth management attempts to manage the capacities assigned to the different VPs. Parts of the network can become under-utilised, and other parts congested. When this occurs, some connections are rejected which could be accepted if the traffic load were better balanced. A connection will be rejected by the network when the capacity reserved for the VP it is to traverse has been used by existing connections so much that there is not enough left for the new connection.

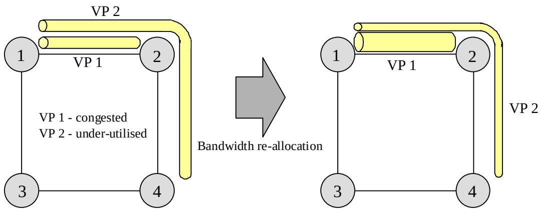

Figure 3: Bandwidth re-allocation or tuning

The idea is to minimise Call Blocking Probability (CBP), i.e. the probability that a call offered has of being rejected due to not enough capacity being available for allocation of the new call. Two actions that are usually taken for the bandwidth management system:

- If in the same link there are congested VPs and under-used VPs, it is possible to reconfigure the bandwidth assigned to each VP in a way that minimises the worst call blocking probability in any given VP. This method can be called bandwidth re-allocation (Figure 3).
- If all the VPs in the link are congested or near congestion and there is not enough unutilised bandwidth capacity for swapping between VPs, then routes as well as capacities are altered to maximise the traffic carried on the network. When this occurs a change in VP network topology is required. When a topology change is required, this can be used to calculate an overall optimal redistribution of the VPs to cope with the actual traffic demand. In other cases, the priority is to minimise the number of re-routed VPs so that the minimum number of connections are affected, in which case the problem becomes a routing one (Figure 4).

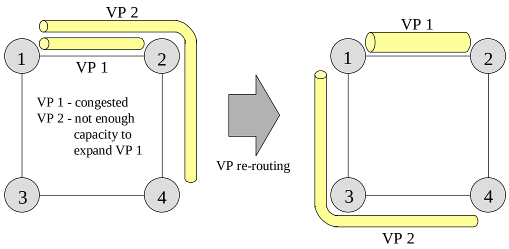

Figure 4: VP re-routing or topology tuning

Networks have to be fault-tolerant. To achieve that, rapid restoration after a failure is required. The ultimate goal is that the customers do not perceive failures. There are two main types of restoration schemes: dynamic and pre-planned. The latter restores effectively and rapidly, but requires more spare resources (Veitch 1996, Yahia and Robach 1997). Pre-planned schemes (Figure 5) are based on pre-assigned backup VP, whereas dynamic schemes are based on flooding algorithms and the search for restoration routes by broadcasting messages after the failure is detected.

In hybrid restoration both schemes are applied, i.e. some priority VPs can be protected with pre-planned schemes and other less-priority ones with dynamic schemes (Yahara and Kawamura 1997, Kawamura and Ohta 1999).

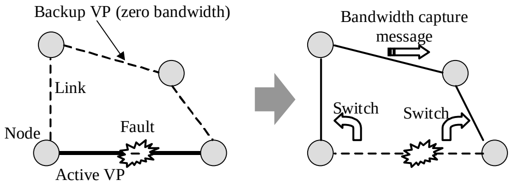

Figure 5: Pre-planned restoration

In this framework, a third problem arises, spare capacity planning. The introduction of hybrid restoration mechanisms is basically due to saving on spare capacity, i.e. the need for spare capacity to protect all the VPs with backup schemes is a very expensive solution. Network providers want economic benefits: as bandwidth is an expensive resource, hence the objective is to minimise the bandwidth reserved for restoration procedures (Xiong and Mason 1999). As mentioned above, hybrid restoration combines pre-planned schemes with dynamic ones in order to save spare bandwidth, which requires good spare capacity planning.

3 SYSTEM ARCHITECTURE

Vila et al. 1999 and Marzo et al. 2000 put forward a system architecture which managed VPs dynamically in ATM networks by using intelligent software agents. The purpose is the integration of conventional ATM management mechanisms and intelligent agent technology. We think that this technology can contribute to improving network management for two main reasons: it is an inherently distributed solution and it introduces into network management 'intelligent' mechanisms which can automate some of the day-to-day tasks of human network managers.

The main objectives of this architecture are the following:

- Maximum integration with the conventional ATM management mechanisms
- Robustness
- Scalability
- Simplicity of the agents.

Robustness is the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environment conditions (IEEE Standard Computer Dictionary 1990). Robustness must be achieved in two ways: the multi-agent system design must itself be robust, and also, in case of multi-agent system failure, the ATM network must continue its normal operations, even though performance may deteriorate.

To achieve these requirements, a complete distributed management system, i.e. without any central vision of the network and without overall data vision, needs to be designed. This introduces another element into the system: how can human network managers manage a system without any overall view of itself? The initial objective is to design a system which can operate without any kind of human intervention; but a future objective will be system interaction with the network operators. Human managers would be able to introduce orders and objectives into the system and the system would be able to give some sort of overview of the network.

The system will be integrated into conventional ATM management, in the sense that it will be placed to help and add new management mechanisms to the Network Resource Management layer, as Figure 6 depicts. The multi-agent system proposed must be transparent for the other ATM mechanisms, principally CAC and routing, with which the system will interact most.

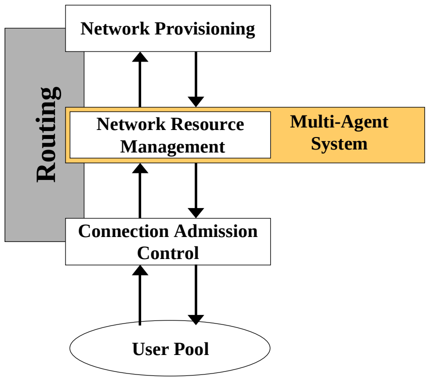

Figure 6: Multi-Agent System placement

The architecture is depicted in Figure 7. Two types of agents are used: Network Monitoring (NM) agents and Network Planning (NP) agents. Of course, all the agents are situated at the nodes where the computing facilities are. There are one NP agent and several NM agents per node.

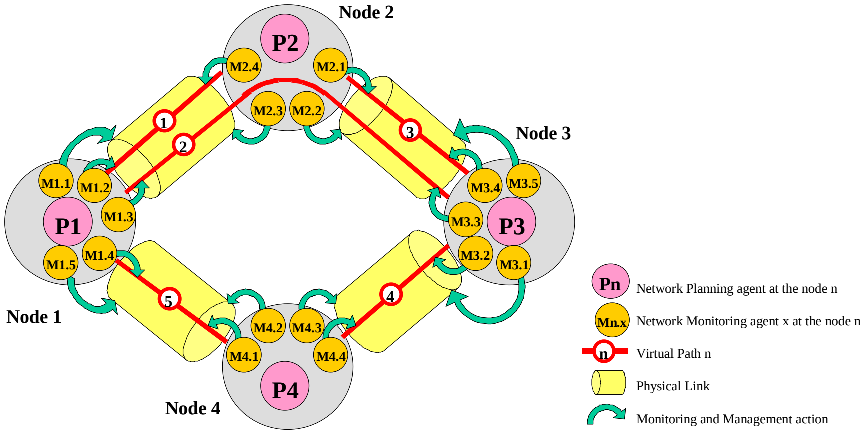

Figure 7: System Architecture

Network Monitoring Agents

The NM agents are supposed to be very simple reactive agents (lightweight processes) with a stimulus/response type of behaviour, whose main task is to monitor and control the resources they are assigned to and react quickly when an event (connection attempts, connection releases, load changes, faults, etc.) occurs. For this purpose each NM agent is responsible for one resource, one VP or one Physical Link (PL), and it has to monitor and control this specific resource. Because of the monitoring task the agents store information as knowledge and statistics, not as bulk data. For the purpose of robustness, two NM agents, one on each end of the VP or PL, manage every resource. These two agents must collaborate. One of the most important tasks of these agents is to implement the restoration mechanisms (which need a fast response). The NM agents are grouped at different priority levels, as they are in charge of different priority VPs.

Network Planning Agents

The NP agents are more deliberative and have the assignment of planning VP topology and bandwidth allocation to achieve good network performance and low call blocking probability. Another function of these agents is to manage spare capacity and plan the backup VPs' need for pre-planned restoration. They also have manage the NM agents (create, destroy, modify, update, consult, etc.), since these represent the resources to be managed (Figure 7). In fact, it is as if the NM agents are a kind of resource wrapper to agentify them, and the NP agents manage the network through the NM agents, i.e. they monitor the NM agents and can modify their rules and goals. No NP agent has a complete distributed overall view of the network, but the NP agents maintain some kind of distributed view by means of co-operation between neighbours, by which each agent has a limited domain view and possesses limited information. However, by pooling their abilities, agents are able to solve problems beyond the capacity of any one single agent.

4 SCALABILITY STUDY

As argued above, scalability has to be studied from two different points of view. First, the number of agents on the network must not grow exponentially when the network grows. Second, the number of messages that the agents interchange in order to manage the network must not increase exponentially when the network grows.

Let us consider a concrete scenario of network growth with a certain number of agents and number of messages between agents. The proposed scenario consists of a fully meshed logical VP network over a fully meshed physical network. When a node is added to the network, it is connected to all the other existing nodes, and the corresponding VPs are created.

Regarding the first problem, how the number of agents grows, it is obvious that NP agents grow linearly with network growth because there is one NP agent per node, which means that we only need consider the number of NM agents. We defined the number of NM agents controlling the physical network and the VPs at two per link (physical or virtual). In a fully meshed network the formula to calculate the number of links (L) is presented in (1), where N means the number of nodes. As there are 2 NM agents controlling each physical link and two more NM agents controlling each VP, the number of NM agents in the sample network is expressed in (2).

$$L = \frac{N(N-1)}{2} \qquad (1)$$

$$NM = 2N(N-1) \qquad (2)$$

This means that the number of NM agents grows quadratically, as against the linear growth of the number of nodes in the network, which is presented graphically in Figure 8. We considered only one VP per physical link, but for each additional VP per physical link, $N(N-1)$ would have to be added to the number of NM agents in (2). This is a large number of agents but is not an exponential increase. Otherwise, NM agents are very simple and lightweight processes.

The second problem, the number of messages that the agents interchange, is more difficult to evaluate since the internal agent details, co-ordination and negotiation mechanisms are still not defined. At any rate, one of the design priorities is to minimise the amount of information that the agents need to interchange, i.e. maximise the autonomy of the agents. To this end, the restriction applied in the Lucent Technologies study (Lucent Technologies 1997) seems very pertinent, i.e. agents at any node can only communicate with agents in the immediately neighbouring nodes (physical neighbours and logical neighbours). For example, in Figure 7, agent P2 can directly negotiate with agents P1 and P3 but not with agent P4 (no direct physical link or VP exists between Node 2 and Node 4), but agent P1 can negotiate with agents P2, P3 and P4.

Two more restrictions, due to the architectural design, have to be considered. First, communications between NP and NM agents take place only inside each node, which means that there is no communication between NM agents in one node and NP agents in other nodes. This kind of communication does not have to be taken into account because it has no effect on the network load, but only on the node load. In this case, moreover, it represents no problem because it is supposed that the intra-node communication is sufficiently powerful (e.g. using shared memory mechanisms). Second, the inter-node communications between NM agents occur only between the pairs of NM agents controlling each physical link or VP. Because of the simplicity of the NM agents these communications are considered to be very simple and sporadic, and to use OAM cells or specially defined cells, which represents a small load at each link.

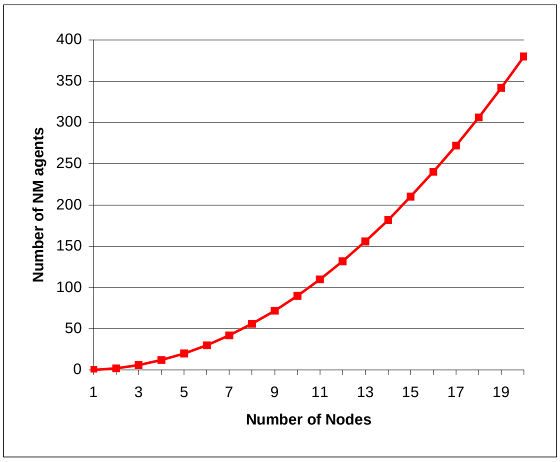

Figure 8: Number of NM Agents – Number of nodes relation

Therefore, we only considered the communications between NP agents, which occur between neighbours, maintain a distributed view of the network, and plan and manage it. If we look again at the sample network discussed in this section (fully meshed VP network), we can see that each node is the neighbour of every other node in the network, and so each NP agent has to communicate with every other NP agent, and each NP agent has to maintain $N - 1$ communications. If we evaluate the number of outgoing messages of each NP agent instead of the communications, we find that N agents sending $N - 1$ messages means that the total number of messages between NP agents is $N(N - 1)$. Again we face a quadratic relationship between the number of nodes and the number of messages required. The crucial point here is the timing of these messages and their size, which is still not known. Note that, in this case, an increase in VPs per physical link does not affect the number of NP agents, and so does not affect the number of messages between the agents, yet may somehow affect the timing of the messages. Also note that in increasing the number of nodes and links we are also increasing the network's capacity to transport information accordingly.

We have looked at the worst case for the physical network, and found that an increase in VPs per physical link does not represent a great change in the number of agents or messages. Moreover, it is easy to see that the real worst case in overall system scalability occurs when, in a not full-meshed physical network, the number of VPs increases without any modification in the physical network. In this situation the number of NM agents increases in the ratio of two more agents to one more VP. These agents, as said above, are lightweight processes and their communication is two-by-two and very sporadic. The number of NP agents does not increase at all, but their messages and their load increase because each NP agent manages more VPs and has more neighbours. In this case it has to be taken into account that there is a limit on the number of VPs per physical link and that using a large amount of VPs between the same nodes is an abnormal use of ATM networks. If this situation occurs, regardless of this misapplication, it is possible to see that the maximum number of neighbours and messages never exceeds the full-meshed situation already discussed.

5 DISTRIBUTED SIMULATION

At present, the system architecture described here has not been fully implemented, as it is a work in progress. A test-bed needs to be designed and implemented, and tests performed. To this end, a first set of experiments is being planned with the following premises:

- The purpose is to evaluate a subset of ATM functions.
- Some ATM functions, which are not part of the system, are required for the evaluation of these new functions.
- There is no need for low-level simulation, i.e. simulation at ATM cell level.
- The system makes sense for a certain number of ATM nodes.
- The idea is to develop a system as similar as possible to a real system controlling a real ATM network.
- Test-bed implementation should be used for another type of experiment with only a few changes.
- An overall goal is to reduce the system's complexity as much as possible.

It can be deduced from these requirements that, in fact, multi-agent architecture has to be implemented as a real system, and it is rather the ATM network that has to be simulated, or in fact emulated. With an emulation of an ATM network and a real system controlling this emulated ATM network, the only thing that has to be strictly simulated is user behaviour, i.e. how the users request/release connections during the time-period.

A set of computers, each one emulating an ATM switch, are used. They run the proposed architecture, with each one controlling its corresponding "emulated ATM node". The computers can be personal computers, the underlying local area network an ethernet, and the operating system Linux, which means that the infrastructure is quite simple and very low-cost. This also enables the agent system to be designed, implemented and to function almost as a real system in a real ATM network. This means that the experiments will be performed using a distributed system, i.e. there will be a distributed simulation.

Following the above requirements, the system is divided into three different and independent subsystems running at each node (the proposed experiment platform is shown in Figure 9):

- The Traffic Event Generator (TEG), which simulates the user's behaviour. Each TEG system is able to simulate new connections to be established or released, and to change the instantaneous load dynamically.

- The ATM Node Model, which emulates the operation of an ATM switch. This system contains all the information on the VP and VC configurations of its node and receives events and notifications from the TEG.
- The Dynamic VP Management System (DVPMS), which adapts the resources on the network, based on the Node configuration information and the inter-agent co-ordination. This system is made up of the two multi-agent subsystems described above (NM-MAS and NP-MAS).

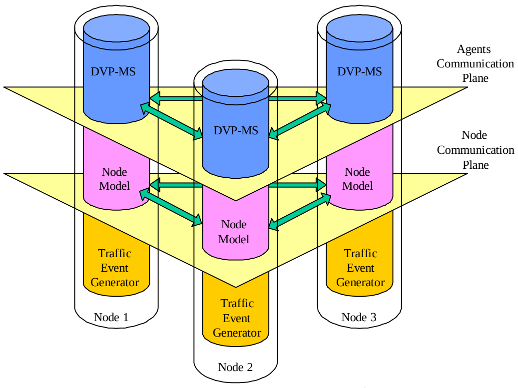

Figure 9: Experimental Platform

Connection Admission Control and Routing functions are carried out by conventional approaches on the Node Communication Plane.

6 CONCLUSIONS AND FUTURE WORK

We have introduced the known problem of the scalability of network management applications, making some comparisons and bringing in some new ideas from the distributed artificial-intelligence world. We have also introduced our proposal for a multi-agent system which can perform dynamic Virtual Path management in ATM networks and have evaluated its scalability. The ideal scalability should be a linear ratio, and the worst case is an exponential ratio. The question is: how good is the quadratic scalability ratio presented here? Given the nature of this kind of distributed system, i.e. multi-agent systems, we think that the ratio is acceptable.

We have also introduced in this paper the distributed simulations we planned for our multi-agent system according to our concrete requirements. Many experiments can be performed using this kind of distributed simulation on our emulated ATM test-bed. We plan to use it also for some experiments to evaluate different charging methods.

7 REFERENCES

Betser, J., 1996, “SNMP/MBD Hybrid Network Management”, IEEE/IFIP NOMS'96 Network Operations and Management Symposium, Japan (April).

Bigham, J., Cuthbert, L.G., Hayzelden, A.L.G., Luo, Z., 1999a, “Multi-Agent System for Network Resource Management”, International Conference on Intelligence in Services and Networks, IS&N'99, Barcelona (April).

Bigham, J., Cuthbert, L.G., Hayzelden, A.L.G., Luo, Z., 1999b, “Flexible Decentralised Control of Connection Admission”, Proceedings of IMPACT’99, Seattle, USA (December).

Black, U., 1994, “Network Management Standards”, 2nd ed., McGraw Hill.

Davison, R.G., Hardwicke, R.G., Cox, M.D.J., 1998, “Applying the agent paradigm to network management”, BT Technology Journal, (July).

Friesen, V.J., Harms, J.J., Wong, J.W., 1996, “Resource Management with Virtual Paths in ATM networks”, IEEE Network, vol 10 no 5, (September/October).

Goldszmidt, G., Yemini, Y., 1998, “Delegated Agents for Network Management”, IEEE Communications Magazine, (March).

Hardwicke, J., Davison, R., 1998, “Software Agents for ATM Performance Management”, IEEE NOMS'98 Network Operations and Management Symposium, New Orleans, USA (February).

Hayzelden, A.L.G., Bigham, J., 1998 “Heterogeneous Multi-Agent Architecture for ATM Virtual Path Network Resource Configuration”, Intelligent Agents for Telecommunications Applications IATA 98, (June).

Hayzelden, A.L.G., Bigham, J., 1999 “Agent Technology in Communications Systems: An Overview”, Knowledge Engineering Review.

Hayzelden, A.L.G., Bigham, J., Luo, Z., 1999, “A Multi-Agent Approach for Distributed Broadband Network Management”, 4th International Conference and Exhibition on Practical Application of Intelligent Agents and Multi-Agents, London, UK, (April).

IEEE Standard Computer Dictionary 1990, “IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries”, Institute of Electrical and Electronics Engineers, New York.

Jogalekar, P, Woodside, M., 1998, “Evaluating the Scalability of Distributed Systems”, Proceedings of International Conference on System Sciences, Hawaii, USA, (January).

Kawamura, R., Ohta, H., 1999, “Architectures for ATM Network Survivability and Their Field Deployment”, IEEE Communications Magazine, (August).

Kyas, O., 1995, “ATM networks”, International Thomson Computer Press.

Larsson S.O., Arvidsson, A., 1997, “A Study of a Distributed Approach for VPC Network Management”, IEEE GLOBECOM’97 Global Telecommunications Conference, Phoenix, USA (November).

Larsson S.O., Arvidsson, A., 1999, “An Adaptive Local Method for VPC Capacity Management”, ITC 16, P. Key and D. Smith (Eds.), Elsevier Science B.V.

Le Boudec, J.Y., 1992, “The Asynchronous Transfer Mode: a tutorial”, Computer Networks and ISDN Systems, vol 24, no 4, (May).

Lee, L.C., Nwana, H.S., Ndumu, D.T., De Wilde, P., 1998, “The stability, scalability and performance of multi-agent systems”, BT Technology Journal, vol 16 no 3, (July).

Lucent Technologies White Paper, 1997, “Proactive Problem Avoidance and QoS Guarantees for Large Heterogeneous Networks”, Lucent Technologies, Rensselaer Polytechnic Institute, and Pennsylvania State University.
URL:http://www.cs.rpi.edu/~kaploww/finalproposal.html

Manley, A., Thomas, C., 1997, “Evolution of TMN Network Object Models for Broadband Management”, IEEE Communications Magazine, (October).

Marzo, J.L., Vilà, P., Fabregat, R., 2000, “ATM network management based on a distributed artificial intelligence architecture”, Accepted in 4th International Conference on Autonomous Agents, AGENTS’2000, Barcelona, Spain (June).

Nwana, H.S., 1996, “Software agents: an overview”, The Knowledge Engineering Review Vol.11 No.3.

Raman, L., 1998, “OSI Systems and Network Management”, IEEE Communications Magazine, (March).

Sato, K.I., Ohta, S., Tokizawa, I., 1990, “Broad-Band ATM Network Architecture Based on Virtual Paths”, IEEE Transactions on Communications, vol 38 no 8, (August).

Sidor, D.J., 1998, “TMN Standards: Satisfying Today's Needs While Preparing for Tomorrow”, IEEE Communications Magazine, (March).

Somers, F., 1996, “HYBRID: Unifying Centralised and Distributed Network Management using Intelligent Agents”, IEEE/IFIP NOMS’96, Network Operations and Management Symposium, (April).

Somers, F., Evans, R., Kerr, D., O’Sullivan, D., 1997, “Scalable low-latency network management using intelligent agents”, ISS’97, World Telecom Congress, (September).

Stallings, W., 1996, “SNMP, SNMPv2 and RMON: Practical Network Management”, Addison-Wesley, ISBN 0-201-63479-1

Stallings, W., 1998, “SNMP and SNMPv2: The Infrastructure for Network Management”, IEEE Communications Magazine, (March).

Sykas, E.D., Vlakos, K.M., Hillyard, M.J., 1991, “Overview of ATM networks: functions and procedures”, Computer Communications, vol 14, no 10, (December).

Veitch, P.A., 1996, PhD thesis “Virtual Path Restoration Techniques for Asynchronous Transfer Mode Networks”, University of Strathclyde.

Vilà, P., Marzo, J.L., Fabregat, R., Harle, D., 1999, “A Multi-Agent Approach to Dynamic Virtual Path Management in ATM Networks”, IMPACT’99, Seattle, USA (December).

Wooldridge, M., Jennings, N.R., 1995, “Intelligent Agents: Theory and Practice”, Knowledge Engineering Review, (January).

Xiong, Y., Mason, L.G., 1999, “Restoration Strategies and Spare Capacity Requirements in Self-Healing ATM Networks”, IEEE/ACM Transactions on networking, (February).

Yahara, T., Kawamura, R., 1997, “Virtual Path self-healing scheme based on multi-reliability ATM network concept”, IEEE GLOBECOM’97, (November).

Yahia, S.B., Robach, C., 1997, “Self-Healing Mechanisms in ATM Networks: The Role of Virtual Path Management Functions”, ICC'97 International Conference in Communications, (June).

Yemini, Y., Goldszmidt, G., Yemini, S., 1991, “Network Management by Delegation”, 2nd International Symposium on Integrated Network Management, Washington DC, (April).